

# Pairwise Alignment - CS262 - Lecture 1 Notes

Scribed by Rong Xu  
xurong@stanford.edu

04/01/2003

## 1 Biology in One Side

- **Molecular Paradigm(Genetic Dogma):**

*DNA* → *RNA* → *Polypeptide*

*DNA* is transcribed into *mRNA* through a process called gene transcription and *mRNA* is translated into *Polypeptide* through protein translation mechanism. Polypeptide then folds into a 3D structure according to its primary amino acid sequence.

- **Evolution Paradigm:**

All organisms come from common ancestor, connected by evolution tree

## 2 High Throughput Biology

Biology changes are driven by new high throughput technologies:

- **DNA Sequencing**

The entire DNA sequences of several organisms, including human, are now available. These are long strings of base pairs (A,C,G,T) containing all the information necessary for an organism's development and life.

- **Sequencing of Expressed Genes(EST sequencing)**

Can find a lot of genes in organism by high throughput technologies

- **Gene Expression: Microarrays**

Different cells have different patterns of gene expression and levels of gene expression are also different. Gene expression determines how different cells function and how one cell differs from the other. Microarray can tell a cell's gene expression pattern and level.

- **Gene Regulation**

Protein binds to DNA to regulate gene expression

### 3 The Goal of Genomics

- **Study organisms at the DNA level**

- Identify "parts" (genes, etc) in DNA
- Figure out "connections" between "parts", how genes interact with each other.

- **Study evolution at the DNA level**

- Compare DNA in different organisms
- Uncover evolutionary history

### 4 The Role of CS in Biology

Because of the shift to high throughput technologies in Biology, the past ten years there has been an explosion of genomics data. Computer science is playing a central role in genomics: DNA sequencing and assembling to piece sequences together, Genome annotation to find genes, repeat families and to discover similarities between sequences of different organism, Microarray analysis which measure and filter the microarray signals, and protein 3D prediction.

The tools that have been used include:

- Alignment algorithm
- multiple alignment algorithms and heuristics such as Gibbs sampling
- Hidden Markov models
- Statistical algorithms

## 5 Why Align Sequence

Complete genome of some organisms have been sequenced. All the genomes are related. Evolution happens at the DNA level through *Sequence Edits*(*substitution, insertion, deletion*) and *Rearrangements*(*translocation, inversion, duplicaion, long range deletion and insertion*).

Mutation in DNA is a natural evolutionay process. They are incorporated into next generation in very different rates. If change happens in non-critical region, it will propagate into subsequent generation. If mutation happens in very important region, changes usually don't propagate into next generation. But some changes are still ok or maybe better, so they will appear in subsequent generation. Sequence conservation implies function. For example, the Interleukin regions in human and mouse are highly conserved. DNA part(gene) which encodes protein is more conserved than the other. Similarity between DNA sequences can be a clue to common evolutionary origin or common function. This motivates the sequence alignment. Sequence alignment can be used to discover functional, structural and evolutionary informtion between the aligned sequences.

## 6 Definition Of Sequence Alignment

Given two strings  $x = x_1x_2 \cdots x_M$ ,  $y = y_1y_2 \cdots y_N$  and a scoring scheme for evaluating matching letters and gap penalty, an alignment is an assignment of gaps to positions  $0 \dots N$  in  $x$  and  $0 \cdots M$  in  $y$ , so as to line up each letter in one sequence with either a letter, or a gap in the other sequence.

*AGGCTATCACCTGACCTCCAGGCCGATGCC*

*TAGCTATCACGACCGCGGTTCGATTTGCCCGAC*

*-AGGCTATCACCTGACCTCCAGGCCGA - -TGCCC - - -*

*TAG \* CTATCAC - -GACCTC - -GGCCGATTTGCCCGAC*

*Optimal alignment* is the optimal pairing of sequences that retains the order of letters in each sequence, perhaps introducing gaps, such that the total score is optimal. It is the same as to find the minimum number of edits that transform one sequence into the other, where the edit operations are insertion, deletion and substitution. The total score of an alignment is the sum of terms for each aligned pair of letters, plus terms for each gap.

The optimal alignment depends on the scoring function used in calculating the total score. Each exact match gets a positive score  $m$ , each mismatch (base substitution) gets a penalty of  $-s$  and each gap (insertion or deletion) gets a penalty of  $-d$ .

$$\text{Score } F = (\# \text{matches}) * m - (\# \text{mismatches}) * s - (\# \text{gaps}) * d$$

The computational complexity of alignment when gaps are introduced is exponential in the length of the sequences being aligned. There are too many possible alignments:  $O(2^{M+N})$  ( $M$  and  $N$  are lengths of two sequences). It is not computationally feasible to enumerate all the paths, even for moderate length.

Since the alignment is additive, *Dynamic programming* can be used in finding optimal alignment.

The score of aligning

$$\begin{array}{c} x_1 \cdots x_M \\ y_1 \cdots y_N \end{array}$$

is additive

Say that  $x_1 \cdots x_i x_{i+1} \cdots x_M$

aligns to  $y_1 \cdots y_j y_{j+1} \cdots y_N$

The two scores add up:

$$F(x_{[1:M]}, y_{[1:N]}) = F(x_{[1:i]}, y_{[1:j]}) + F(x_{[i+1:M]}, y_{[j+1:N]})$$

Suppose we wish to align

$$\begin{array}{c} x_1 \cdots x_M \\ y_1 \cdots y_N \end{array}$$

let  $F(i, j)$  = optimal score of aligning

$$\begin{array}{c} x_1 \cdots x_i \\ y_1 \cdots y_j \end{array}$$

Notice that there are three possible cases:

1.  $x_i$  aligns to  $y_j$

$$\begin{array}{c} x_1 \cdots x_{i-1} \ x_i \\ y_1 \cdots y_{j-1} \ y_j \end{array}$$

$$F(i, j) = F(i - 1, j - 1) + m, \text{ if } x_i = y_j$$

$$F(i, j) = F(i - 1, j - 1) - s, \text{ if } x_i \neq y_j$$

2.  $x_i$  aligns to a gap

$$\begin{array}{r} x_1 \cdots x_{i-1} \quad x_i \\ y_1 \cdots y_j \quad - \end{array}$$

$$F(i, j) = F(i - 1, j) - d$$

3.  $y_j$  aligns to a gap

$$\begin{array}{r} x_1 \cdots x_i \quad - \\ y_1 \cdots y_{j-1} \quad y_j \end{array}$$

$$F(i, j) = F(i, j - 1) - d$$

How do we know which case is correct?

*Inductive assumption:*

If  $F(i, j - 1)$ ,  $F(i - 1, j)$ ,  $F(i - 1, j - 1)$  are optimal, Then

$$F(i, j) = \max \begin{cases} F(i - 1, j - 1) + s(x_i, y_j) \\ F(i - 1, j) - d \\ F(i, j - 1) - d \end{cases}$$

Where  $s(x_i, y_j) = m$ , if  $x_i = y_j$ ,  $-s$  if not

## 7 Alignment Algorithm

The Needleman-Wunsch Algorithm is a general algorithm for sequence comparison and it finds the best GLOBAL alignment of any two sequences. It involves an iterative matrix method of calculation according to above scoring function.

- **The Needleman-Wunsch Matrix**

All possible pairs of residues(DNA bases or protein amino acids)- one from each sequence - are represented in a 2-dimensional array. The sequences are written across the top and down the left side of the matrix, except that an extra row(row #0) and column(column #0) are added to allow the alignment to begin with a gap of any length in either sequence. The gap rows are filled with penalty scores for gaps

of increasing lengths. Maximum possible values are calculated for all other boxes below and to the right of the top row and left column using the above scoring functions. All possible alignments are represented by pathways through this matrix. Each cell is the maximum possible score for an alignment ending at that point. For each cell, look at all possible pathways back to the beginning of the sequence(allowing gaps) and give that cell the value of the maximum scoring pathway.

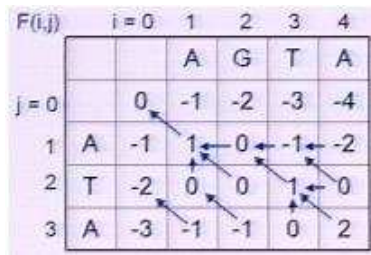


Figure 1: Filled Needleman-Wunsch Matrix and Traceback Pointers

Every nondecreasing path from  $(0,0)$  to  $(M,N)$  corresponds to an global alignment of the two sequences.

- **The Needleman-Wunsch Algorithm**

1. Initialization

$$F(0,0) = 0$$

$$F(0,i) = -i * d$$

$$F(j,0) = -j * d$$

2. Main Iteration

For each  $i = 1 \dots M$

For each  $j = 1 \dots N$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j), \text{ case 1} \\ F(i-1, j) - d, \text{ case 2} \\ F(i, j-1) - d, \text{ case 3} \end{cases}$$

$$Ptr(i,j) = \begin{cases} \text{DIAG}, \text{ if case 1} \\ \text{LEFT}, \text{ if case 2} \\ \text{UP}, \text{ if case 3} \end{cases}$$

### 3. Termination

$F(M, N)$  is the optimal score, and from  $Ptr(M, N)$ , we can trace back the optimal alignment

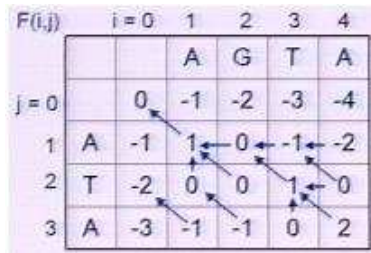


Figure 2: The Optimal Alignment

The optimal global alignment is:

*AGTA*

*A - TA*

### 4. Performance

- . Time:  $O(NM)$  (We need to fill out the whole matrix)
- . Space:  $O(NM)$  (We need a matrix to store all the traceback pointers)

#### • A Variant Of The Basic Algorithm and Overlap Detection

If entire sequences are supposed to be similar, then end gap should be penalized. If sequences are of very different lengths or they are known to overlap one another, it is OK to have an unlimited number of gaps in the beginning and end:

-----CTATCACCTGACCTCCAGGTCGATGCCCTTCCGGC  
 GCGAGTTCATCTATCAC--CACCTC--GGTCG-----

Then we don't want to penalize gaps in the ends.

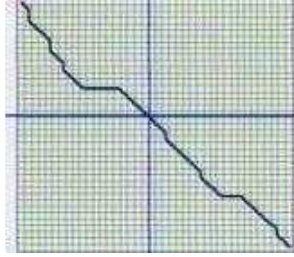


Figure 3: Global Alignment, End Gap Penalized



Figure 4: Overlap Alignment, End Gap Not Penalized

To avoid penalizing end gaps, a few changes need to be made from the Needleman-Wunsch Algorithm:

1. Initialization

For all  $i, j$ :

$$F(0, i) = 0$$

$$F(j, 0) = 0$$

2. Termination

$$F_{OPT} = \max \begin{cases} \max F(N, i) , i = 1 \dots M \\ \max F(j, M) , j = 1 \dots N \end{cases}$$